VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Vizualizace grafů

Graph visualisation

2016 Bc. Michal Ondra

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Michal Ondra**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: Vizualizace grafů
Graph Visualisation

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem práce je implementace vybraného algoritmu pro vizualizaci grafů a experimentální ověření této implementace.

1. Přehled současných metod vizualizace grafů.
2. Výběr a návrh vlastní metody vizualizace grafů.
3. Implementace a experimenty s navrženou metodou.
4. Porovnání s jinými přístupy.
5. Zhodnocení dosažených výsledků.
6. Závěr.

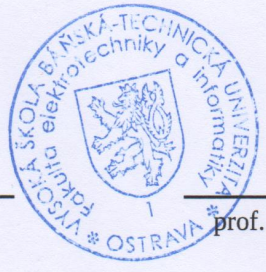
Seznam doporučené odborné literatury:

[1] Shixia Liu, Weiwei Cui, Yingcai Wu, Mengchen Liu. A survey on information visualization: recent advances and challenges,
[2] http://link.springer.com/article/10.1007%2Fs00371-013-0892-3

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **prof. RNDr. Václav Snášel, CSc.**

Datum zadání: 01.09.2015
Datum odevzdání: 29.04.2016



doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 29. dubna 2016 ..... Michal Ondra

Děkuji panu prof. RNDr. Václavu Snášelovi, CSc. za to, že se ujal vedení mé diplomové práce, jeho cenné rady a připomínky.

Abstrakt

Cílem této práce je navrhnout a analyzovat metodu pro vizualizaci grafů sociálních sítí. Práce představuje algoritmus, který slučuje vrcholy grafu podle jejich stupně ve vzestupném pořadí. Výsledkem tak je zjednodušený graf sítě. Poté navazuje metoda, která filtruje hrany na základě jejich betweenness centralit. Také je vytvořena aplikace pro vizualizaci výsledného grafu. Navržená metoda je testována na datech reálných sociálních sítí.

Klíčová slova: graf, sociální síť, analýza sociálních sítí

Abstract

The aim of this thesis is to design and analyse method for graph visualisation of social networks. The thesis introduces an algorithm, that merges verticies of graph based on the their degree. The result is a simplified graph of the network. In addition, a method, that filters edges based on their betweenness centrality is used. There is also implemented an application for visualisation of the output graph. Designed method is tested on data of real social networks.

Key Words: graph, social network, social network analysis

Obsah

Seznam použitých zkratek a symbolů

DBLP	–	Digital Bibliography & Library Project
SQL	–	Structured Query Language
WPF	–	Windows Presentation Foundation
KONECT	–	the Koblenz Network Collection

Seznam obrázků

Seznam tabulek

1 Úvod

V současné době jsou online sociální sítě velmi populární. Tyto sítě spojují mnoho lidí, které by se za normálních okolností nikdy neznaly a pomáhá jim udržovat kontakt. S nárůstem počtu lidí na těchto sítích je však stále těžší vyhledávat data nebo je analyzovat ať už z důvodu nedostatku paměti nebo kvůli příliš dlouhé době zpracování. Důležitá část analýzy nejen sítí, ale i dat obecně je právě jejich vizualizace. Pro tyto účely musejí být zaváděny nové metodiky, které zvládají pracovat i s tak rozsáhlými daty.

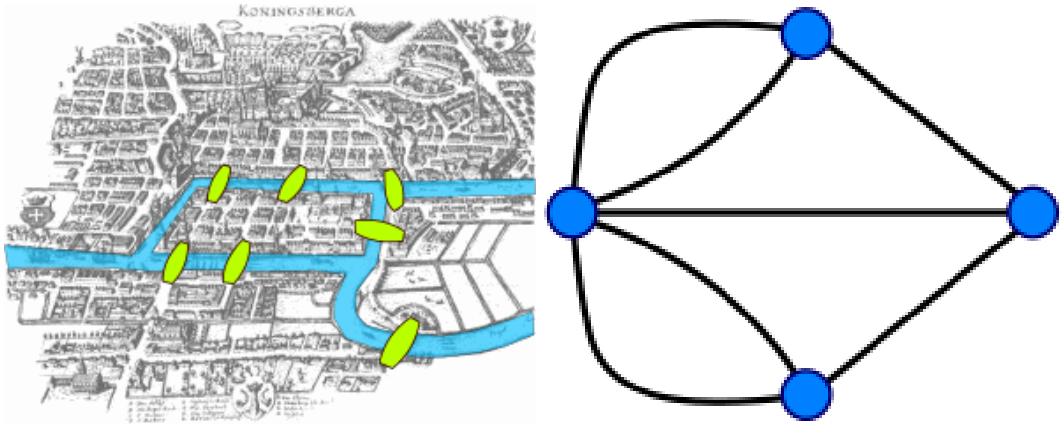
V rámci této práce se zamýšlím nad tímto problémem a vytvářím vlastní řešení. Také jsou představeny některé existující způsoby a jejich porovnání. V první části je definován graf, pojmy a vlastnosti s ním spojené. Dále se zaměřuji na oblast rozsáhlých sítí, zejména sociálních a uvádím některé důležité vlastnosti, které se berou v úvahu při jejich analýze. Následuje rozbor procesu vizualizace grafů a některých metod, které se používají pro vizualizaci velkých sítí. V další části se věnuji návrhu a implementačním detailům navržené metody a aplikace, která slouží pro vizualizaci grafů za použití navržené metody. Poslední část je věnována experimentům a analýze této metody při použití na reálných sociálních sítích.

2 Úvod do teorie grafů

V této kapitole je definován graf a uvedeny základní pojmy a symboly, které jsou dále používány v této práci.

2.1 Definice grafu, rozdělení

Vznik teorie grafů je datován do 18. století. V té době bylo publikováno řešení tzv. problému sedmi mostů Königsbergu (dnes Kalingrad) [17]. V tomto ruském městě protéká řeka Pregole, na které bylo v té době vystavěno 7 mostů. Úloha spočívala v otázce, zda lze přejít po všech sedmi mostech právě jednou a zároveň se vrátit do místa, ve kterém se začíná. Tento problém vyřešil matematik Leonhard Euler převedením a tedy abstrakcí úlohy na strukturu množin vrcholů a hran, viz obr. 1 [17]. Snadno pak dokázal své tvrzení, že přejít po každém mostě právě jednou a vrátit se na původní místo je nemožné. Struktura, kterou použil pro zjednodušení úlohy byla později pojmenována právě jako graf. V té době také vznikla již zmíněná studie teorie grafů, která se věnuje těmto strukturám.



(a) Nákres problému sedmi mostů (b) Odpovídající graf

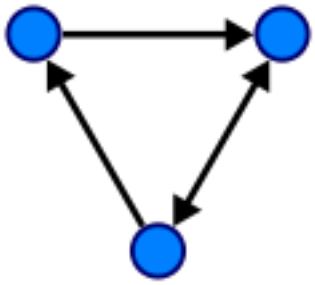
Obrázek 1: Převedení úlohy 7 mostů Königsbergu na příslušný graf.

Grafem se rozumí obecná struktura vrcholů a relací mezi nimi [7]. Formálně je graf G definován jako uspořádaná dvojice množin vrcholů V a hran E:

Definice 1 *$G = (V, E)$, kde V je neprázdná množina vrcholů $V = \{v_1, v_2, \cdots, v_n\}$ a $E = \{e_1, e_2, \cdots, e_m\}$ je množina hran. Hrana $e_a = (v_b, v_c) \in E$ je definována jako dvojce vrcholů z množiny V.*

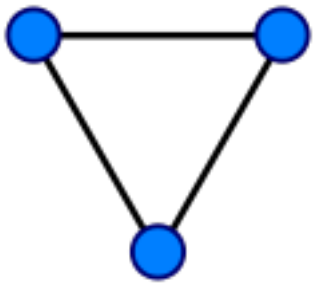
Velikost grafu určuje počet jeho vrcholů $|V|$. Grafy se nejčastěji znázorňují pomocí diagramů tak, že vrcholy z množiny V představují body nebo geometrické objekty a každá hrana z množiny

E je pak spojení mezi nimi, obvykle čára nebo křivka. Grafy se dělí na orientované a neorientované. Orientované grafy obsahují hrany, které jsou pouze jednosměrné. To znamená, že například hrana $e = (v_a, v_b)$ spojuje vrchol v_a s vrcholem v_b, avšak ne naopak. Tyto hrany se pak podle směru zakreslují jako šipky. Mezi dvojicí vrcholů může existovat dvojce hran s opačným směrem. V tomto případě se někdy zobrazují tyto hrany jako jedna s šipkou na obou koncích spojení.



Obrázek 2: Orientovaný graf

Neorientované grafy jsou naopak grafy, které obsahují pouze obousměrné hrany. Pokud se tak lze dostat z jednoho vrcholu do jiného, jde to i naopak.



Obrázek 3: Neorientovaný graf

V teorii grafů jsou také zavedeny tzv. smíšené grafy $G = (V, E, A)$, definované jako množina vrcholů V, neorientovaných hran E a orientovaných hran A. Tyto grafy jsou tak kombinacích výše uvedených grafů a mohou obsahovat jak orientované hrany, tak i neorientované.

2.2 Vlastnosti a pojmy

Výše uvedená definice je velice obecná. Dovoluje totiž zavádět různé typy grafů. Například existují grafy s nekonečně velkými množinami vrcholů a hran–tzv. nekonečné grafy. Ty však nebudou dále uvažovány.

Obecně není omezen počet hran mezi dvojicemi vrcholů. Neorientovaný graf $G = (V, E)$, kde $e_1 = (v_1, v_2), e_2 = (v_3, v_4), e_1, e_2 \in Eav_1 = v_3av_2 = v_4$ a pokud jsou povoleny hrany $e = (v_a, v_b)$, kde $v_a = v_b$ (tzv. smyčky) se nazývá multigraf. Naopak neorientovaný graf, který neobsahuje smyčky ani rovnoběžné hrany je jednoduchý graf. Vážené grafy mohou mít ohodnocené hrany

číslem $w \in \mathbb{R}$. Toto číslo se nazývá váha hrany. Nejčastěji určuje délku nebo cenu příslušné hrany.

Každý vrchol grafu může mít sousedy, kterými jsou myšleny vrcholy spojené hranou s tímto vrcholem. Stupeň vrcholu $deg(v)$ udává počet hran incidentů vrcholu v, tedy počet hran, které spojují tento vrchol s jinými vrcholy, případně se sebou samým, viz 1. Smyčky jsou počítány dvakrát. Maximální stupeň libovolného vrcholu grafu, kde nejsou povoleny smyčky ani paralelní hrany, je $deg_{max}(v) = |V| - 1$.

$$deg(v) = \sum_{u} A(v,u) + \sum A(v,v) \tag{1}$$

Definice 2 *Cesta v grafu* $\sigma_{v_s,v_t} = (e_0, e_1, \ldots, e_n)$ *označuje posloupnost hran takovou, že* $e_0 = \{v_s, v_i\}$, $e_m = \{v_p, v_q\}$, $e_{m+1} = \{v_q, v_r\}$, $e_n = \{v_j, v_t\}$ *a kde se žádná z hran nevyskytuje více než jednou.*

Definice 3 *Tah je cesta, ve které se mohou opakovat vrcholy.*

Definice 4 *Sled je cesta, ve které se mohou opakovat vrcholy i hrany.*

Definice 5 *Kružnice označuje cestu, kdy* $v_0 = v_n$.

Graf, který neobsahuje žádné kružnice se nazývá acyklický graf nebo také les.

Definice 6 *Acyklický graf, který je souvislý je strom. Vrcholy se stupněm* $deg(v) = 1$ *se nazývají listy.*

Definice 7 *Délka cesty mezi dvěma vrcholy* $d(v_i, v_j)$ *se rovná počtu hran, kterou prochází, resp. pro vážené grafy* $\sum_{e \in E} w(e)$

Definice 8 *Vzdálenost mezi dvěma vrcholy* $d(v_i, v_j)$ *označuje délku nejkratší cesty mezi těmito vrcholy.*

Definice 9 *Excentricita* $E(v)$ *vrcholu v je největší možná délka mezi vrcholem v a jiným libovolným vrcholem* v_0.

Jinými slovy se jedná o vzdálenost mezi vrcholem v a vrcholem v_0, který je od něj nejdál.

Definice 10 *Poloměr grafu r udává minimální excentricitu v grafu* $G(E,V)$ *nebo také* $r = minE(v)$, *kde* $v \in V$. *Průměr grafu d označuje maximální excentricitu v grafu* $G(E,V)$ *nebo také* $d = maxE(v)$, *kde* $v \in V$.

Definice 11 *Graf je souvislý, pokud mezi všemi dvojicemi jeho vrcholů existuje cesta.*

U orientovaných grafů se rozlišuje souvislost na slabou a silnou. Orientovaný graf $G = (V, E)$ je silně souvislý, pokud $\forall v_a, v_b \in V, \exists e_a, e_b \in E : e_a = (v_a, v_b), e_b = (v_b, v_a)$. Pro slabě souvislý graf naopak platí $\exists v_a, v_b \in V, \exists e_a \in E, \exists e_b \notin E : e_a = (v_a, v_b), e_b = (v_b, v_a)$.

Definice 12 *Komponentou grafu se myslí jeho část, pro kterou platí, že je souvislý.*

Dá se tedy říct, že souvislý graf obsahuje právě jednu komponentu. Pro komponenty orientovaných grafů platí stejné podmínky slabé a silné souvislosti.

Definice 13 *Graf je úplný, pokud každý jeho vrchol v je spojen hranou s ostatními vrcholy.*

Úplný graf má poloměr r, průměr d i excentricitu E(v) všech vrcholů = 1. Pokud má úplný graf n-vrcholů, nazývá se n-úplným.

Definice 14 *Pokud $G = (V, E)$ a $G' = (V', E')$ tak, že $V' \subseteq V$, $E' \subseteq E$ a $\forall e = (v_1, v_2), e \in E'$ a $v_1, v_2 \in V'$, pak G' je podgrafem grafu G.*

Například pro graf $G = (v_1, v_2, v_3, v_1, v_2, v_2, v_3, v_1, v_3))$ jsou možné podgrafy $G'_1 = (v_1, 0)$, $G'_2 = (v_1, v_2, v_1, v_2)$, nebo samotný G.

2.3 Sociální sítě

Síť je definována jako jakékoli uskupení vzájemně propojených entit. Sociálními sítěmi se označuje specifická skupina sítí, které zaznamenávají vztahy mezi osobami nebo jinými sociálními prvky, např. organizacemi nebo státy. Sítě jsou reprezentovány grafy. Vrcholy těchto grafů představují jednotlivé osoby, a pokud jsou dva vrcholy spojeny hranou, znamená to, že mezi těmito osobami existuje nějaký vztah.

Tyto sítě mají využití zejména pro analýzu nebo grafické znázorňovaní struktury uživatelů sociálních sítí a jejich vztahů. Některé základní kategorie online sociálních sítí jsou [15]:

- Všeobecné sítě pro navazovaní vztahů mezi uživateli. Příklady těchto sítí jsou Facebook [1] a Twitter [2].
- Služby, které jsou určeny pro sdílení multimediálního obsahu.
- Profesionální sociální sítě. Tyto sítě jsou určeny pro skupiny lidí se stejným profesním zaměřením. Uživatelé mohou navazovat mezi sebou vztahy a připojovat se ke různým skupinám. Největší profesionální síť je LinkedIn [3].
- Vzdělávací sítě, které jsou určeny pro spolupráci studentů na akademických projektech, výzkumech nebo ke komunikaci mezi studenty a lektory.

[1] facebook.com
[2] twitter.com
[3] linkedin.com

Uživatelé mohou k těmto sítím přistupovat pomocí webových služeb. Vytvořením účtu se připojí k síti a mohou navazovat vztahy navzájem s ostatními členy sítě.

Mimo online sítě existují i sociální sítě představující lidi v reálném světě. Příkladem jsou např. sítě spoluautorství. Členové těchto sítí jsou autoři vědeckých publikací a jejich vazby znázorňují to, že spolu někdy spolupracovali.

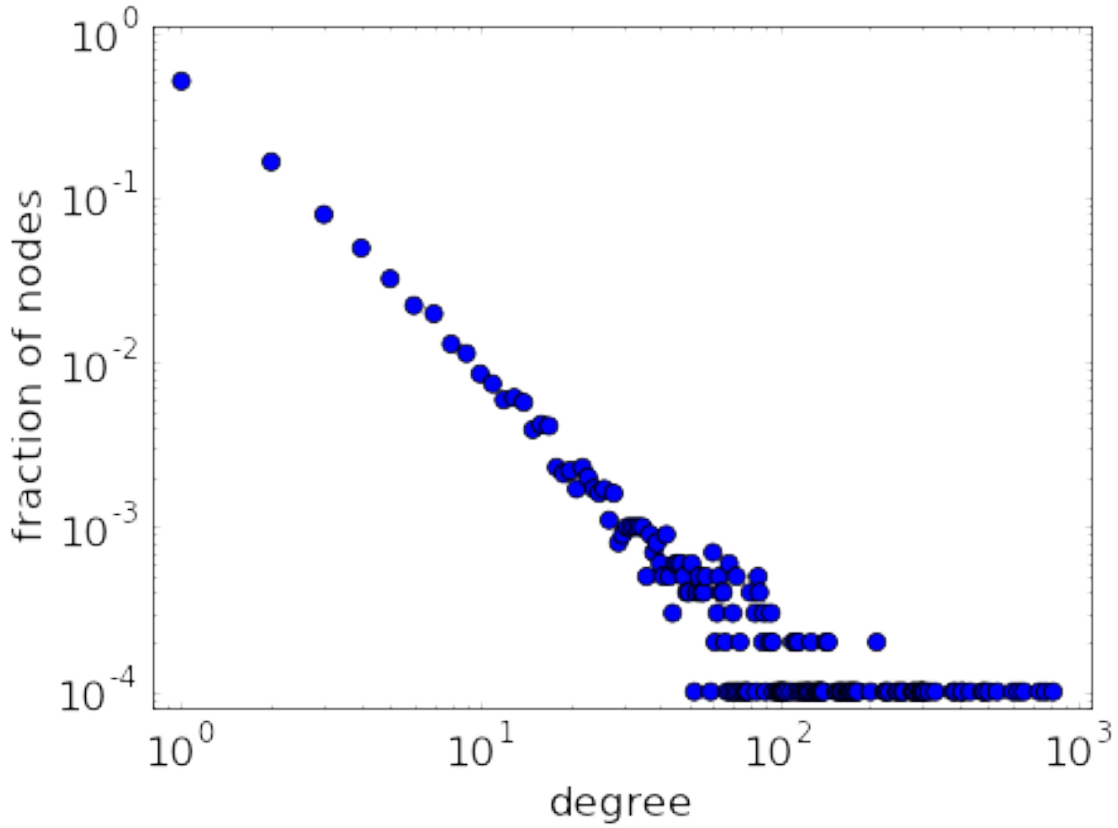
3 Vlastnosti sociálních sítí

Sociální grafy mohou být orientované i neorientované. Například graf sociální sítě Facebook, kde hrana představuje přátelství mezi dvěma osobami, je grafem neorientovaným. Graf sociální sítě Twitter, kde jedna osoba sleduje druhou, avšak ne nutně opačně, je grafem orientovaným.

Sociální sítě sdílejí několik podobných vlastností, jelikož všechny představují interakci mezi osobami. Ukázalo se, že většina vrcholů sociálních sítí má stupeň v řádu jednotek a pouze malý počet vrcholů má desítky až stovky sousedů [18]. Také mají vzhledem ke své velikosti velmi malý průměr v řádech jednotek. Tento jev se nazývá fenomén malého světa. V roce 1967 provedl sociální psycholog Stanley Milgram experiment. Rozeslal stovky dopisů náhodně vybraným osobám v USA s tím, že příjemci tyto dopisy poslaly lidem, které znají, dokud tento dopis nedorazil zpět k odesílateli. Na základě výsledků tohoto experimentu vznikla teorie nazývaná šest stupňů odloučení, která udává, že mezi libovolnými dvěma osobami na planetě existuje cesta průměrné délky 6 [13]. Ačkoliv bylo vysloveno mnoho námitek, které znevažují Milgramův experiment, při analyzování mnoha sociálních sítí se ukázalo, že tato teorie má hodně blízko ke skutečnosti. Zvlášť v posledním století mnoho lidí navazují kontakty s lidmi po celém světě, což vede k propojování předtím oddělených společností.

3.1 Bezškálové sítě

Sociální sítě spadají do kategorie tzv. bezškálových sítí. Bezškálové sítě jsou sítě, jejichž distribuce vrcholů splňuje tzv. mocninné rozdělení (viz obr. 4) [16]. Taková síť obsahuje malý počet vrcholů s relativně velkým stupněm (tzv. centrální vrcholy). Sousedi centrálních vrcholů mívají již menší stupeň a se vzdáleností od těchto vrcholů se stupeň snižuje na minimum. Bezškálové sítě také jsou sítě malého světa, tedy s malým průměrem, který se nemění s přidáváním nových vrcholů. Další vlastností bezškálových sítí je vysoký shlukovací koeficient, což znamená, že vrcholy mají tendenci se zcela propojovat v malých shlucích. U sociálních sítích se tyto shluky také nazývají komunity.



Obrázek 4: Mocninné rozdělení vrcholů

3.2 Centralita vrcholů

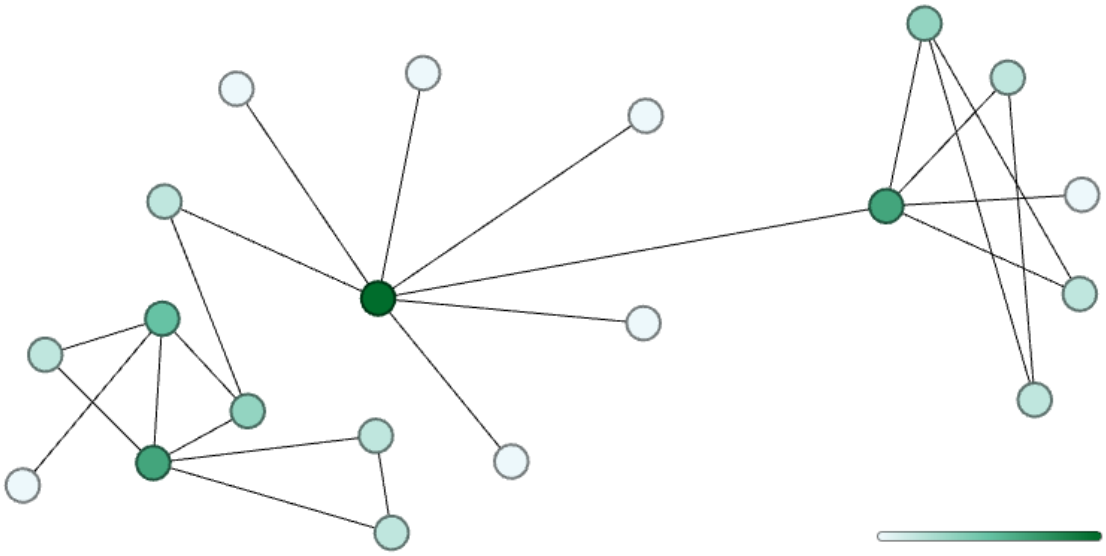
Vrcholy mají svou důležitost v rámci struktury sítě. Tato vlastnost se nazývá centralita. Existují různé míry centrality založené na stupni vrcholu nebo na vzdálenosti mezi vrcholy [3]. Následovně představím tři nejdůležitější pro analýzu grafů sociálních sítí.

3.2.1 Degree centralita

První a nejjednodušší míra centrality udává, že čím má vrchol větší stupeň, tím je také důležitější a má větší vliv na své sousedy. U orientovaných grafů se rozlišuje degree centralita na in-degree pro hrany vstupující do vrcholu a out-degree pro hrany, které z vrcholu vychází. Následující vzorec slouží pro výpočet degree centrality C_D vrcholu v:

$$C_D(v) = deg(v), v \ \in V \tag{2}$$

na obrázku č. 5 je zobrazen ukázkový graf. Barva vrcholů indikuje degree centralitu. Čím je vrchol tmavší, tím větší je hodnota degree centrality.



Obrázek 5: Degree centralita

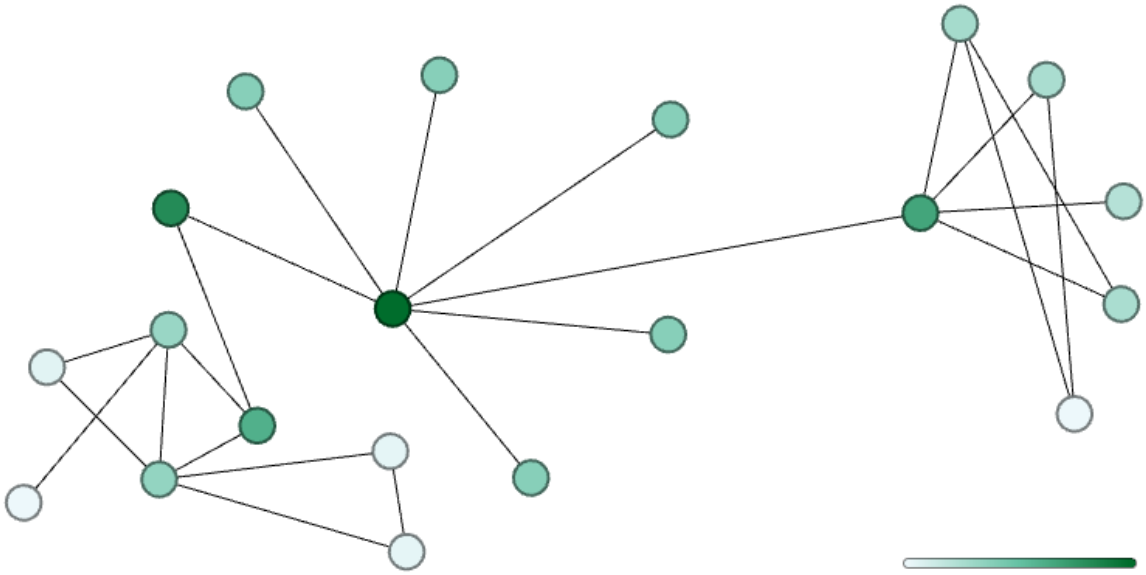
Z hlediska sociálních sítí degree centralita určuje jedince, kteří jsou nejvíce známí nebo naopak znají ostatní. Tuto míru centrality však ovlivňují jen sousedi vrcholu, proto spadá do kategorie lokálních centralit [3].

3.2.2 Closeness centralita

Closeness centralita C_C je založená na nejkratších cestách mezi vrcholem a ostatními vrcholy v grafu, viz vzorec č. 3. Pokud není graf souvislý, počítají se pouze cesty k vrcholům, které jsou ve stejné komponentě a poté se výsledek vynásobí počtem vrcholů v dané komponentě. Closeness centralita udává pro jedince sociální sítě míru toho, jak blízko má k ostatním členům sítě [3].

$$C_C(v) = \sum_{u \in V} \frac{1}{d(v,u)} \tag{3}$$

Opět uvádím ukázkový graf na obr. 6 se stejným značením velikostí centralit vrcholů.



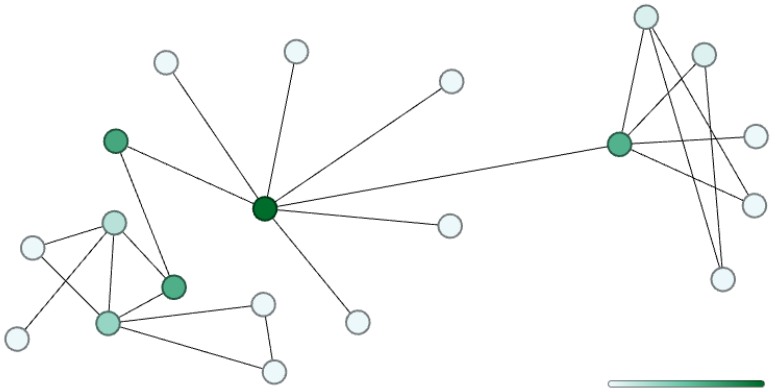
Obrázek 6: Closeness centrality

3.2.3 Betweenness centralita

Další míra udává centralitu vrcholu podle toho, jak moc jsou na něm ostatní vrcholy sítě závislé, pokud chtějí mezi sebou vázat vazby. Betweenness centralita je dána poměrem počtu nejkratších cest mezi všemi dvojicemi vrcholů, které procházejí právě tímto vrcholem, a všem těmto cestám, viz vzorec č. 4.

$$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}, kde\ s, t, v\ \in V. \tag{4}$$

Následující obrázek ukazuje betweenness centralitu vrcholů ukázkového grafu.



Obrázek 7: Betweenness centralita vrcholů

3.3 Hustota grafu

Další důležitou vlastností je hustota grafu. Maximální počet hran v neorientovaném grafu vzhledem k počtu jeho vrcholů je

$$|E|_{max} = \frac{|V| * (|V| - 1)}{2} \tag{5}$$

Pokud je graf orientovaný, maximální počet hran je $|V| * (|V| - 1)$. Hustota grafu udává poměr mezi počtem hran v grafu a maximálním možným, tedy

$$hustota = \frac{|E|}{|E_{max}} \tag{6}$$

Obecně mají sociální sítě malou hustotu. Grafy s vysokou hustotou se nazývají husté a naopak, grafy s nízkou hustotou se nazývají řídké.

3.4 Komunity

V životě obvykle nemáme dostatek času ani energie k tomu poznávat tisíce nebo milióny lidí (záleží, o jak velké síti se bavíme). Lidé mají tendenci se shlukovat spíše do malých skupin. Pokud jeden člověk zná jiného a ten zase zná dalšího, je relativně vysoká pravděpodobnost, že ho první zmíněný člověk zná také. V sociálních sítích tak vznikají navzájem provázané skupiny lidí, které se nazývají komunity. Komunity jsou definovány jako podgrafy sítě s velmi vysokou hustotou. Zároveň jsou spojení mezi komunitami řídké. Komunity jsou zpravidla malé – řádově jednotky až desítky vrcholů.

4 Reprezentace grafů

V této části jsou popsány způsoby reprezentace grafů v paměti.

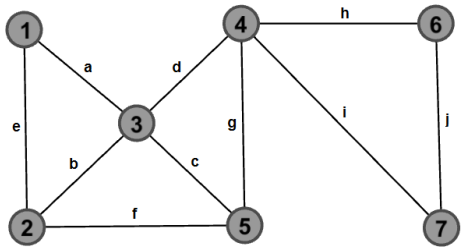
4.1 Matice sousednosti a incidencí

S příchodem počítačů začal být zájem o to zkoumat grafy a jejich možné aplikace v počítačovém světě. Nejjednodušším způsobem, jak reprezentovat grafové struktury dat je matice sousednosti. Mějme neorientovaný graf $G = \{V, E\}$. Můžeme pak zavést matici M o rozměrech $|V| \times |V|$ [7]. Čísla této matice mohou nabývat hodnot 0 nebo 1. Pokud je hodnota rovna 1 na pozici M_{ab}, pak existuje hrana $e = v_a, v_b$ v grafu G . Alternativně, pokud je hodnota 0, pak taková hrana neexistuje. Je zřejmé, že hodnoty na diagonále matice představují smyčky.. Pokud by jsme uvažovali vážený graf G, pak by čísla matice M nabývali reálných hodnot. Pokud by však tyto hodnoty mohly nabývat váhy 0, je nutné vytvořit druhou matici, která obsahuje váhy na příslušných pozicích v matici a se samotnou maticí sousednosti se bude pracovat stejně, jako když reprezentuje nevážený graf [7].

Matice sousednosti je obvykle implementovaná jako dvojrozměrné pole. Výhoda tohoto přístupu je snadná implementace a přehlednost. Časová složitost přístupu k jakékoli hraně je konstantní $\mathcal{O}(1)$ za předpokladu, že známe indexy obou vrcholů, které hrana spojuje. Velkou nevýhodou je prostorová složitost $\mathcal{O}(|V|^2)$ uložení této matice. Je zřejmé, že matice sousednosti neorientovaných grafů je symetrická. Protože na směru hran v neorientovaném grafu nezáleží, každá hrana je v této matici reprezentovaná dvakrát. Použití matice sousednosti je vhodné pouze pro malé grafy s ohledem na velikost v paměti nebo pro velmi husté orientované grafy. Dále, přidávání nebo odebírání nových vrcholů je náročná operace s časovou složitostí $\mathcal{O}(|V|^2)$, protože je nutné překopírovat prvky do nového pole. Naopak, přidávání nebo odebírání hran v matici sousednosti má konstantní časovou složitost $\mathcal{O}(1)$, případně $\mathcal{O}(2)$ pro neorientované grafy.

Obdobný přístup je matice incidencí N. Tato matice má rozměr $|V| \times |E|$ [7]. Řádky této matice představují vrcholy, sloupce zase hrany. Hodnota $N_{i,j}$ tak nabývá 1, pokud i-tý vrchol spojuje j-tá hrana s jiným vrcholem, nebo hodnotu 0 pokud nikoli. Pokud je graf orientovaný, hodnoty nabývají 1 pro hrany vystupující z vrcholu a -1 pro hrany, které do něj vstupují. Reprezentace vážených grafů samotnou matici incidencí není možná. Každá hrana může spojovat právě dva vrcholy, což znamená, že v každém sloupci budou dvě hodnoty, zbytek jsou nulové hodnoty. Počet nenulových hodnot v matici incidencí je tak rovno $|E| * 2$. Prostorová složitost uložení matice incidencí je $\mathcal{O}(|V| \cdot |E|)$. Přidávání nebo odebírání vrcholů má prostorovou i časovou složitost $\mathcal{O}(|V| \cdot |E|)$, protože je nutné vytvořit nové pole a překopírovat zde všechny hodnoty.

Pro snadnější pochopení je uveden příklad převodu grafu na obrázku č. 8 na příslušné matice sousednosti i incidencí (obr. č. 9).



Obrázek 8: Ukázkový graf

$$\begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} \end{array}$$

(a) Matice sousednosti

$$\begin{array}{c} \begin{matrix} a & b & c & d & e & f & g & h & i & j \end{matrix} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} \end{array}$$

(b) Matice incidencí

Obrázek 9: Matice sousednosti a incidencí

4.2 Seznam sousednosti

Dalším způsobem reprezentace grafů se nazývá seznam sousednosti. Každý vrchol uchovává kolekci odkazů na své sousedy [7]. V případě, že se jedná o vážený graf, je vhodné použít hash mapu, kde klíč bude sousední vrchol a hodnota bude váha. V paměti však budou reference na každý vrchol dvakrát, pokud uvažujeme neorientovaný graf. Nevýhodou je složitější přístup– zatímco u incidenční matice pro přístup k hraně objektu stačilo znát index obou vrcholů, které spojuje, nejlepší způsob nalezení požadované hrany u objektového přístupu je procházení kolekce sousedů vrcholu.

5 Algoritmy pro určení vlastností grafů

Základem analýzy sítí je výpočet jejich vlastností. K tomu je nutné vybrat správný algoritmus. V této kapitole uvádím některé nejpoužívanější netriviální algoritmy a jejich zhodnocení.

5.1 Určení počtu komponent grafu

První vlastností, která nás bude zajímat je souvislost grafu. Následujícími algoritmy lze ověřovat, zda je graf souvislý nebo určit počet jeho komponent v případě, že souvislý není. Pro tento účel se používají algoritmy pro průchod grafem, kterými jsou prohledávání do hloubky a prohledávání do šířky. Oba jsou velice podobné se stejnou časovou složitostí $\mathcal{O}(m+n)$ a prostorovou složitostí $\mathcal{O}(m)$, kde m je počet vrcholů grafu a n počet hran.

Prohledávání do šířky využívá datové struktury fronta. Při cestování grafem z určitého zdrojového vrcholu se do fronty ukládají sousední vrcholy, které ještě v daný okamžik nebyly navštíveny. Následným odebráním z fronty se tyto vrcholy samy stanou zdrojovými vrcholy. Zároveň se vede seznam již navštívených vrcholů. Pokud se fronta vyprázdní, znamená to, že byly navštíveny všechny vrcholy, které se nacházejí v aktuální komponentě. Algoritmus pak může najít vrchol, který ještě nebyl navštíven a postup opakovat. Pokud seznam navštívených vrcholů obsahuje všechny vrcholy v grafu, algoritmus je u konce.

Prohledávání do hloubky naopak využívá zásobník. Na počátku se zvolí zdrojový vrchol a algoritmus se přesune ihned na první sousední vrchol, který nalezne. Zároveň se do zásobníku ukládá předchozí vrchol, které navštívil. Postup se opakuje, dokud se nedostane k vrcholu, který nemá žádné nenavštívené sousedy. V tomto případě se přesune zpět na poslední vrchol uložený v zásobníku. Opět se přemístí na nenavštívený sousední vrchol nebo na další vrchol v zásobníku a pokračuje. Pokud se zásobník vyprázdní, algoritmus navštívil všechny vrcholy aktuální komponenty.

Pokud je graf orientovaný, slabou souvislost lze určit jednoduše tak, že se graf považuje za neorientovaný. Pro ověření silné souvislosti se používá např. Tarjanův algoritmus, jehož časová složitost je také $\mathcal{O}(n+m)$.

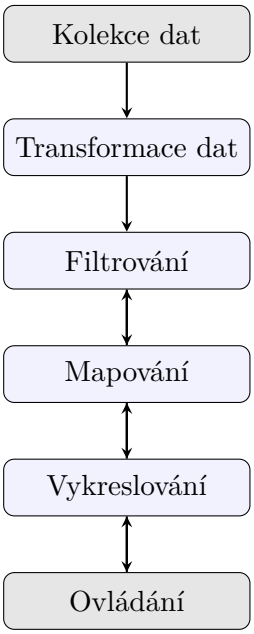
5.2 Betweeness centrality

Výpočet betweenness centrality je časově velice náročné. Tradiční přístup je založen na nalezení všech nejkratších cest mezi každou dvojicí vrcholů a vrcholům, které se nacházejí na těchto cestách, se přičtou hodnoty betweenness centrality. Algoritmus se dá mírně urychlit eliminováním dvojic vrcholů, mezi kterými neexistuje cesta nebo rozdělením grafu na podgrafy představující komponenty a výpočet tak paralelizovat pro každou komponentu zvlášť. Za nejrychlejší přístup je považován Brandesův algoritmus [1], jehož časová náročnost je $\mathcal{O}(n \cdot m)$, kde n je počet vrcholů grafu a m počet hran.

Pro velké grafy se zavedly metody pro výpočet aproximované betweenness centrality, které samozřejmě dávají zkreslené výsledky, avšak pro většinu aplikací jsou dostačující. Výhodou těchto algoritmů je ten, že dokáží vypočítat hodnoty v rozumném čase. Pro bezškálové sítě lze betweenness centralitu aproximovat hledáním cest pouze mezi vrcholy s největší degree centralitou . Počet těchto vrcholů je $c * \log n$, kde n je počet vrcholů grafu a c konstanta zpřesňující výpočet.

6 Vizualizace grafů

Jak již bylo několikrát ukázáno, grafy se nejčastěji vizualizují pomocí geometrických tvarů. Transformace vstupních dat na výsledný graf však není úplně triviální. Celý proces je znázorněn diagramem na obrázku č. 10 [11]. Fáze, které převádějí vstupní data na výsledný obrázek grafu jsou rozepsány v následujících podkapitolách.



Obrázek 10: Obecné schéma procesu vizualizace grafů

6.1 Vstupní data

Vstupem celého procesu vizualizace grafů jsou určitá data. Pokud se zaměříme pouze na analýzu nebo vizualizaci sociálních sítí, data jsou obvykle dodávána formou grafových souborů, pro které byly vytvořeny desítky standardizovaných formátů. Další možností jsou grafové databáze, které se v poslední době stávají čím dál populárnější. Jedná se o speciální NoSQL databáze, navržené pro data nabývající grafové struktury. Oproti relačním databázím nabízejí možnost uchovávat semistrukturovaná data. Možností úložišť pro data sociálních sítí je ale i více.

6.2 Transformace dat

Načtení dat do paměti. Pokud jsou data nestrukturována, jsou převedená na strukturovaná např. pomocí shlukovacích metod . Také dochází k jejich redukci, pokud jsou data příliš velká pro načtení do paměti [11].

6.3 Filtrování

V této fázi dochází k další redukci dat. Důležité je zvolit vhodnou metodu tak, aby výsledný graf byl reprezentativní, ale zároveň aby nebyl příliš velký. Současné metody pro vykreslování si umí poradit i s velkými grafy, problémem je, že vrcholů a hran takových grafů je příliš mnoho, aby dávaly nějaký smysl. Často nás spíše zajímají vzory v síti, než její celek – v případě sociálních sítí to jsou komunity, kostra grafu, komponenty sítě atp [8].

6.4 Mapování

Mapování na geometrické objekty. Hranám i vrcholům jsou přiřazeny atributy, jako je barva, velikost, pozice nebo textový popis. Pozice objektů v prostoru je počítána pomocí tzv. layout algoritmů.

Dále mohou být aplikovány algoritmy, které eliminují překrývání vrcholů jinými vrcholy nebo edge bundling algoritmus.

6.5 Vykreslování

Převedení geometrických dat na výsledný obrázek nebo vykreslení na plátno.

6.6 Ovládání

Uživatel může graf prozkoumávat, měnit pozici vrcholů nebo jinak upravovat. Pro vizualizaci velkých grafů je tento krok někdy vynechán a výsledkem vizualizace je jen statický obrázek grafu.

Každá změna grafu uživatelem se promítá zpět. Pokud například dojde ke změně pozice vrcholu, aplikace se vrátí do fáze mapování, kde se přepočítá nová poloha vrcholu a následně se znovu vykreslí.

7 Existující algoritmy

V posledních letech se výzkumy v oblasti analýzy sociálních sítí zaměřují na velké sítě. Vizualizace, která je důležitá pro jejich analýzu už není tak triviální záležitostí, jelikož nastávají problémy s vykreslováním tak velkých grafů v reálném čase. S vyšším počtem prvků také narůstá náročnost na paměť počítače.

Nejjednodušším řešením je vizualizace podgrafu sítě. Uživatel si může postupně volit vrcholy a jejich sousedy, které chce zobrazit. Výhodou tohoto přístupu je, že lze postupně načítat jednotlivé prvky na principu návrhového vzoru lazy loading a vizualizace je tak rychlá a efektivní. Tento přístup však má omezené využití, protože neřeší problém vizualizace sítě jako celku, který může být příliš velký. Proto byly zavedeny nové metody toho, jak získat zjednodušený graf sítě, s kterým by již bylo možné manipulovat. Zjednodušeným grafem je myšlen graf, který má podobné vlastnosti a uspořádání prvků, jako graf původní sítě.

7.0.1 Random sampling metody

Způsoby, jak získat reprezentativní podsíť jsou např. metody vzorkování grafu na základě náhodného výběru vrcholů a hran nebo na principu náhodného průchodu grafem. Způsobů vzorkování bylo navrženo několik, např. Random walk nebo Forest fire. Uvádí se, že 15% vzorků z původního grafu jsou obvykle dostatečné pro zachování statických i dynamických vlastností původní sítě [9]. Na druhou stranu je chybovost poměrně velká, proto by se vzorkování mělo zvážit s ohledem na situaci. Random sampling metody například nezachovávají souvislost grafu nebo případné komunitní uskupení.

7.0.2 Filtrování hran

Dalším způsobem, jak získat podsíť je filtrování hran [5]. Filtrování probíhá na základě betweenness centralit jednotlivých hran, které určují, jak je hrana důležitá v síti. Odstraňují se hrany, které mají nejmenší betweenness centralitu. Zároveň se kontroluje, zda nedošlo k rozpojení některých komponent grafu. Počet hran k odstranění volí uživatel. Tato metoda je vhodná pro grafy s velkým počtem hran, jelikož počet vrcholů zůstává zachován.

7.0.3 Edge Bundling

Mimo algoritmy pro rozvrstvení vrcholů grafů existují i layout algoritmy pro hrany. Jedním z těchto algoritmů jsou tzv. Edge bundling algoritmy, mezi které patří například Force-Directed Edge Bundling [4] nebo Multilevel Agglomerative Edge Bundling [2], která je navržena pro velké sítě. Tyto metody jsou založeny na slučování hran, které jsou blízko u sebe a sdílejí stejnou cestu, podobně jako se svazují síťové kabely nebo elektrické vodiče v místech, kde mají společnou cestu a na konci těchto cest se zase oddělují. Takto se docílí podstatně lepší viditelnosti grafu.

Edge bundling metody se aplikují ve fázi mapování, kdy mají vrcholy vypočítány pozice v prostoru pomocí některého layout algoritmu. Dají se tak snadno používat v kombinaci s metodami pro filtrování vrcholů nebo hran. Zatímco metoda Force-Directed Edge Bundling je vhodná pouze pro grafy s menším počtem hran, Multilevel Agglomerative Edge Bundling se dá používat i pro komplexnější sítě v reálném čase. Hrany, které reprezentují úsečky spojující dva vrcholy, se přepočítávají na křivky. Hrany jsou převedeny na vektory o 4 dimenzích, respektive 6, pokud je graf vykreslován do 3D prostoru. Pomocí klasifikační metody k-nejbližších sousedů se pro každou hranu najde k nejbližších hran, které mají nejkratší euklidovskou vzdálenost. Autoři metody Multilevel Agglomerative Edge Bundling uvádějí, že časová složitost této metody je v nejlepším případě $\mathcal{O}(km \cdot \log m)$, kde m je rovno počtu hran vstupního grafu, a v nejhorším případě $\mathcal{O}(m^2)$, k čemuž však v praxi nedochází.

7.0.4 Alternativní metody

Existují také alternativní způsoby vizualizace sítí, které nevykreslují jejich graf tradičním způsobem. Jedním z těchto přístupů je vizualizace matice sousednosti [14]. Tato metoda je vhodná pro velmi velké sítě s velkou hustotou. Také je vhodná pro vizualizaci shlukování vrcholů sítě.

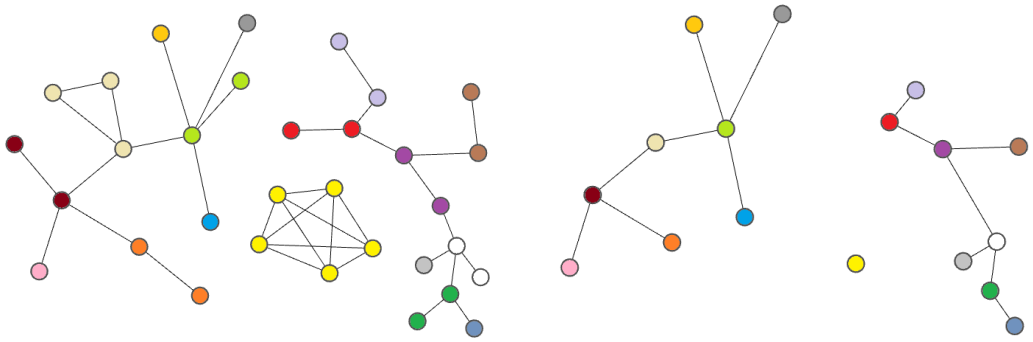
8 Návrh vlastní metody

Navržený algoritmus slučuje vrcholy, které mají nejnižší degree centralitu, s vrcholy v určité vzdálenosti cesty od tohoto vrcholu. Tímto sloučením vzniká zjednodušený graf. Postup vytváření zjednodušeného grafu je následující:

Vstup: graf $G = (V, E)$, konstanta τ
Výstup: zjednodušený graf G_S

1: přiřazení vrcholů do shluku $S = (V', E'), V' \subseteq V, E' \subseteq E$.
2: vytvoření zjednodušeného grafu $G_S = (V_S, E_S)$, takového, kde každý vrchol představuje jeden shluk vrcholů grafu původního a hrany představují spojení mezi těmito shluky
3: **for all** $v \in V_S$ **do**
4: výpočet průměrné betweenness centrality hran $s \in S$ a přiřazení k odpovídajícímu vrcholu v
5: **end for**

Příklad slučování vrcholů je zobrazen na obr. 11. Na obrázku 11a je graf s 31 vrcholy. Barva vrcholů znázorňuje shluk, do kterého vrcholy patří. Na obrázku 11b je pak zjednodušený graf.



(a) Vstupní graf (b) Zjednodušený graf

Obrázek 11: Ukázkový příklad slučování vrcholů pro $\tau = 1$.

Sloučením vrcholů lze dosáhnout k jejich podstatné redukci, avšak v případě velkých grafů může být problémem množství hran, které vrcholy spojují. Proto svou metodu doplňuji o metodu filtrování hran na základě jejich betweenness centralit [5].

Vstup: graf $G_S = (V_S, E_S)$, konstanta k udávající požadovaný počet hran k odstranění
Výstup: podgraf $G' = (V_S, E')$ takový, že $E' \subseteq E_S$

1: přiřazení indexu komponenty vrcholům, do které patří
2: výpočet aproximované betweenness centrality hran B_C
3: odstranění požadovaného počtu hran s nejnižší B_C
4: zpětné doplňování hran, které rozpojily komponenty

Vzhledem k tomu, že se zjednodušený graf liší od původního grafu, je nutné určit centrální vrcholy jiným způsobem, než podle stupně vrcholů v původním grafu. Na výběr jsou 3 možnosti - podle počtu vrcholů shluku, který tvoří vrchol zjednodušeného grafu, podle průměrné betweenness centrality hran shluku nebo podle součtu stupňů vrcholů ve shluku, které propojují tyto vrcholy s vrcholy jiného shluku. Tyto kritéria jsou porovnávány v kapitole 10.

9 Implementace

V této kapitole je popsán návrh a implementace navržené metody a aplikace pro vizualizaci grafů.

9.1 Analýza a návrh aplikace

Cílem implementace je realizovat navrženou metodu pro zjednodušení vstupního grafu a následná vizualizace. Při návrhu byly definovány tyto funkční požadavky:

- Aplikace bude schopna načíst vstupní graf ze souboru.
- Aplikace bude schopna zobrazit výsledný graf.
- Aplikace bude schopna uložit výsledný graf do souboru.
- Uživatel bude mít možnost zadávat parametry navržené metody.
- Uživatel bude mít možnost vybírat mezi několika algoritmy pro vykreslování.
- Aplikace bude hlásit stav výpočtu a dobu trvání.

Kvalitativní požadavky jsou následující:

- Aplikace musí být responzivní během výpočtu.
- Vstupy aplikace musí být ošetřeny vůči neplatným vstupům.
- Aplikace bude rozdělena na komponenty.

Aplikace byla vytvářena v jazyce C#. Pro implementaci uživatelského rozhraní byla zvolena technologie WPF.

9.2 Architektura aplikace

Aplikace je rozdělena na dvě komponenty. První komponentou je knihovna GraphVisualisation, která obsahuje všechny třídy implementace navržené metody vizualizace. Druhá komponenta Visualiser představuje Aplikaci pro testování a vizualizaci výsledného grafu. Při implementaci byly použity tyto externí knihovny:

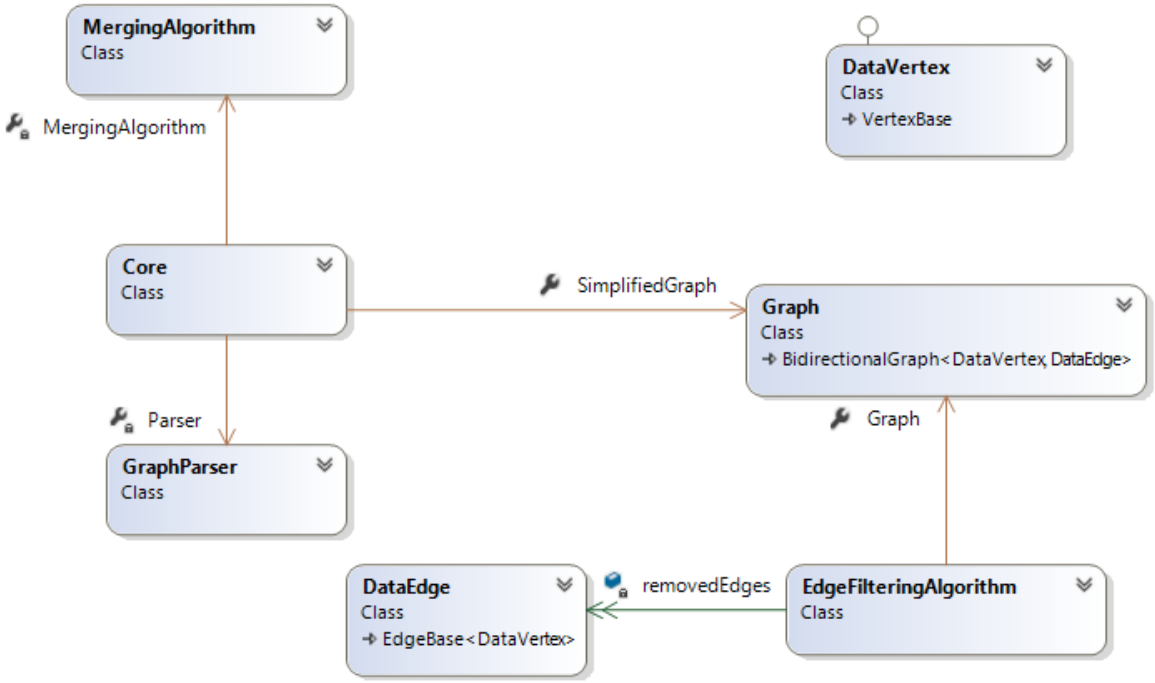
- GraphX [4]- poskytuje WPF ovládací prvky pro vizualizaci grafů a implementaci různých algoritmů pro vykreslování grafů.
- Quickgraph [5]- poskytuje implementaci grafových struktur.

[4] https://graphx.codeplex.com/
[5] https://quickgraph.codeplex.com

Knihovna Quickgraph poskytuje implementaci několika typů grafů. Z možností byl vybrán Bidirectional graph, který může reprezentovat orientované i neorientované grafy. Knihovna GraphVisualisation, obsahuje následující třídy:

- Core - klíčová třída, která zaobaluje celý proces vytvoření zjednodušeného grafu.
- DataEdge - třída, která představuje hranu grafu. Obsahuje mimo jiné vlastnosti nutné pro výpočet betweenness centrality hran.
- DataVertex - tato třída představuje vrchol grafu.
- Graph - reprezentace grafu. Rozšiřuje třídu BidirectionalGraph knihovny QuickGraph.
- EdgeFilteringAlgorithm - implementace metody pro filtrování hran grafu. Také obsahuje implementaci Brandesova algoritmu pro výpočet betweenness centrality a algoritmus prohledávání do šířky, který určuje počet komponent grafu a vrcholům přiřazuje index komponenty, ve které se nalézají.
- GraphParser - třída, která zajišťuje načtení grafu ze souboru a ukládání.
- MergingAlgorithm - třída obsahující implementaci navrženého algoritmu pro vytváření zjednodušeného grafu.

Třídní diagram knihovny GraphVisualisation je na obr. 12.



Obrázek 12: Diagram knihovny GraphVisualisation

9.3 Postup

Celý proces vytváření zjednodušeného grafu zaobaluje třída Core. Provádění probíhá na samostatném vlákně, aby výpočet neblokoval hlavní vlákno aplikace. Pro hlášení stavu v průběhu provádění metody byl vytvořený event ProgressChanged, který předává tyto zprávy objektům, které jsou registrovány. Tímto je také zajištěna nezávislost knihovny na uživatelském rozhraní. Konstruktor třídy Core přebírá jako vstupní parametry cestu k souboru obsahující graf a parametry metody. Metoda Run provádí celý proces vytváření zjednodušeného grafu. Vstupní parametry jsou:

- Cesta - cesta k souboru, který obsahuje graf.
- Délka cesty - konstanta τ.
- Hran - procentuální hodnota udávající, kolik má obsahovat zjednodušený graf hran poměrem k celkovému počtu.
- C - konstanta udávající přesnost betweenness centrality. Vyšší hodnota zajišťuje přesnější výpočet, avšak značně ho zpomaluje.
- Obnovovací krok - tato hodnota udává ve fázi, která zajišťuje zachování souvislosti komponent při filtrování hran, počet hran, které se mají zpětně doplnit, než se souvislost ověří. Vyšší hodnota zrychluje výpočet za cenu toho, že graf bude obsahovat více hran, než je požadováno.
- Váha vrcholů - udává kritérium, podle kterého se určují centrální vrcholy zjednodušeného grafu pro výpočet betweenness centrality hran.
- Orientace - zaškrtávací tlačítko, které určuje, zda je graf orientovaný nebo ne.

9.3.1 Transformace dat

V této fázi je načítán vstupní soubor, který obsahuje graf. Vstupní soubor je ve formátu seznamu hran (přípona edges), který obsahuje dvojce čísel, mezi kterými je mezera. Čísla představují indexy vrcholů grafu. Tyto dvojce představují jednotlivé hrany grafu. Soubor může obsahovat komentáře, které začínají znakem % nebo #. Vstupní graf může být orientovaný i neorientovaný, avšak nejsou povoleny smyčky. V případě orientováného grafu je uvažována pouze out-degree centralita. Načtení grafu ze souboru zajišťuje třída GraphParser metodou ParseGraph, která přijímá jako parametr název vstupního souboru. Data nejsou v této fázi upravována.

9.3.2 Slučování vrcholů

Vrcholy grafu jsou dále sloučeny se svými sousedy. Uživatel si může zvolit konstantu $\tau \in \mathbb{N}$. Na základě degree centrality se prochází vrcholy, které mají tuto hodnotu nejnižší. Algoritmus

vybere vrcholy, které jsou od aktuálního vrcholu ve vzdálenosti τ a menší. Těm poté přiřadí index shluku, do kterého patří. Poté, co je všem vrcholům v grafu přiřazen index shluku, se vytváří zjednodušený graf tak, že vrchol tohoto grafu je shluk vrcholů v grafu původním. Hrany původního grafu propojující vrcholy, které patří do různých shluků jsou i hrany, které tyto shluky propojují. Paralelní hrany však nejsou povoleny.

V této části je vytvořen zjednodušený graf $G_S = (V_S, E_S)$. Následuje příprava na filtrování hran. Oproti původnímu přístupu, kdy se bere v potaz stupeň vrcholu, se vypočítává betweenness centralita hran od vrcholů s největší váhou. Výpočet této váhy lze zvolit ze tří možností. Důvod těchto možností je čistě experimentální.

- První z možností je přiřazení váhy podle počtu vrcholů ve shluku. Vrchol zjednodušeného grafu, který představuje shluk vrcholů grafu původního můžeme chápat jako podgraf, který tvoří vrcholy v tomto shluku a hrany mezi nimi.

- Druhá možnost je určení váhy na základě stupně shluku. Vrcholy tvořící jeden shluk mají v původním grafu vazby i na okolní vrcholy, tedy ty, které tvoří jiný shluk. Součtem těchto vazeb dostaneme stupeň shluku. Tato hodnota se váže na původní graf, protože vrchol zjednodušeného grafu, který představuje shluk má jiný stupeň než je tato hodnota.

- Poslední možností určení váhy vrcholu je výpočet průměrné betweenness centrality hran. Shluk vrcholů a hran mezi nimi tvoří podgraf. Výpočtem průměrné betweenness centrality hran určíme váhu odpovídajícího vrcholu zjednodušeného grafu. Rychlost výpočtu vah všech vrcholů zjednodušeného grafu je závislá na zvolené hodnotě τ. Čím větší je hodnota tohoto parametru, tím více vrcholů jsou v jednotlivých shlucích, což zpomaluje výpočet betweenness centralit hran. Protože žádný vrchol původního grafu nemůže patřit do více, než jednoho shluku, můžeme jednoduše původní graf rozdělit na jednotlivé podgrafy a paralelizovat tak výpočet pro každý shluk. Pro implementaci byla použita paralelní smyčka forearch, která automatizuje vytváření a spouštění vláken. Zároveň také určuje optimální počet těchto vláken, přičemž maximální počet nebyl v implementaci omezen.

9.3.3 Filtrování hran

Nyní se aplikuje metoda filtrující hrany grafu. Počet hran k odstranění volí uživatel. Nejprve probíhá algoritmus pro určení souvislosti zjednodušeného grafu. Všem vrcholům sítě se přiřadí index komponenty, do které vrcholy patří. Po odstranění hran tak lze snadno zjistit, zda nedošlo k rozpojení jedné komponenty na více komponent. Pro tento účel byl použit algoritmus procházení do šířky. Postup je následující:

Vstup: graf $G = (V, E)$
Výstup: počet komponent grafu G

1: početKomponent $\leftarrow 0$
2: **while** existují neobjevené vrcholy **do**

```
3:    root ← první nalezený neobjevený vrchol
4:    root.objeven ← true
5:    root.indexKomponenty ← početKomponent
6:    fronta.enqueue ← root
7:    while fronta není prázdná do
8:      v ← fronta.dequeue
9:      for all sousedi vrcholu v do
10:       if soused nebyl dosud objeven then
11:         soused.objeven ← true
12:         soused.indexKomponenty ← početKomponent
13:         fronta.enqueue ← soused
14:       end if
15:     end for
16:   end while
17:   početKomponent ← početKomponent +1
18: end while
```

Algoritmus předpokládá, že vrcholy mají vlastnost objeven. Pokud ne, lze použít pole boolean prvků pro tyto účely. Vrcholy také mají vlastnost udávající index komponenty, ve které se nacházejí. Jak již bylo zmíněno, časová složitost tohoto algoritmu je $\mathcal{O}(|V| + |E|)$, což je dostatečné i pro velké grafy. Počet hran se může rovnat až $|V| * |V-1|$ v případě orientovaných úplných grafů, avšak u grafů sociálních sítí je počet podstatně menší, jak bylo řečeno v podkapitole 3.3.

Následuje výpočet betweenness centralit hran grafu. V podkapitole 3.2.3 byla uvedena definice betweenness centrality. Definice se však vztahovala pouze na vrcholy. V tomto případě je nutná betweenness centralita hran. Upravením rovnice dostaneme betweenness centralitu hran:

$$C_B(e) = \sum_{s \neq t} \frac{\sigma_{st}(e)}{\sigma_{st}}, kde\ e \in E, s, t \in V. \tag{7}$$

Betweenness centralita udává důležitost hran na základě toho, kolikrát se vyskytují na nejkratších cestách mezi všemi dvojicemi vrcholů grafu G. Výpočet nejkratších cest pro grafy s vysokým počtem vrcholů je však velmi časově náročné. Proto se počítá pouze přibližná betweenness centralita tak, že se počítají vzdálenosti pouze mezi malým počtem centrálních vrcholů s nejvyššími váhami. Počet vybraných centrálních vrcholů je uveden jako $c * \log n$, kde $n = |V|$. Konstantu c volí uživatel. Čím větší je tato hodnota, tím přesnější je výpočet betweenness centralit, avšak doba algoritmu se značně zvyšuje. Pro menší sítě je minimální počet vybraných centrálních vrcholů 50. Pro výpočet betweenness centrality byla zvolena metoda navržená Ulrikem Brandesem [1] s časovou složitostí \⇳, kde n je počet zvolených centrálních vrcholů a $m = |E|$.

Posledním krokem je samotné odstraňování hran. Zde vzniká problém, protože odstranění hrany může porušit souvislost komponenty. Protože graf zjednodušený graf G_S může obsahovat vysoký počet hran, které jsou určeny k odstraní, nelze ověřovat zachování souvislosti po každém

odstranění hrany. Proto se používá zásobník, do kterého se odstraněné hrany ukládají. Po odstranění požadovaného počtu hran se ověří souvislost. Vrcholy mají vlastnosti udávající index komponenty, do které patřily před odstraňováním hran a index komponenty, do které patří po tomto procesu. Pokud tyto hodnoty nejsou stejné, došlo k rozpojení komponent grafu. V tomto případě se provede zpětné přidávání hran ze zásobníku, tedy v pořadí podle nejnižší betweenness centrity hran. Pokud se nalezne hrana, která spojuje různé vrcholy v jiných komponentách grafu $G'_S = (V_S, E'_S), E' \subseteq E_S$, které jsou v G_S v jedné komponentě, hrana se přidá zpět. Postup se opakuje, dokud se takto nespojí všechny rozpojené komponenty. Nutno dodat, že výsledný počet hran se může lišit od požadovaného počtu.

9.3.4 Mapování a vykreslování

Po zpracování může uživatel zvolit z nabídky layout algoritmů vrcholů i hran a algoritmus pro odstranění vzájemného překrytí hran. Dále je poskytnut seznam, ve kterém lze zvolit, zda zobrazit celý zjednodušený graf, nebo jednotlivé shluky původního grafu. Zvolený graf je předán jako parametr pro vykreslování. Vykreslování probíhá pomocí ovládacího prvku GraphArea. Jelikož výpočet probíhá asynchronně, hlavní vlákno aplikace není zahlcené během vykreslování. Uživatel také může uložit zjednodušený graf do externího souboru ve formátu GraphViz (přípona gv).

10 Experimenty

V této kapitole jsou popsány provedené experimenty a uvedeny jejich výsledky. Účelem experimentů je analýza navržené metody a zjištění toho, na které typy grafů je tato metoda vhodná. Klíčové statistické údaje jsou doba výpočtu pro grafy o různých velikostech a vlastnosti grafů jak původních, tak i jejich zjednodušených verzí.

Analyzována je navržená metoda slučování vrcholů i metoda pro filtrování hran. Jako vstupní data byly vybrány série grafů o různých velikostech. Tabulka č. 1 znázorňuje vybrané sítě a jejich charakteristiky pro experimenty.

Název	Počet vrcholů	Počet hran	Typ
Dolphins	62	159	neorientovaný graf
Síť citací DBLP	12 591	49 728	orientovaný graf
High Energy Physics - Theory	9 877	25 998	neorientovaný graf
Hamsterster friendships	1 258	12 534	neorientovaný graf

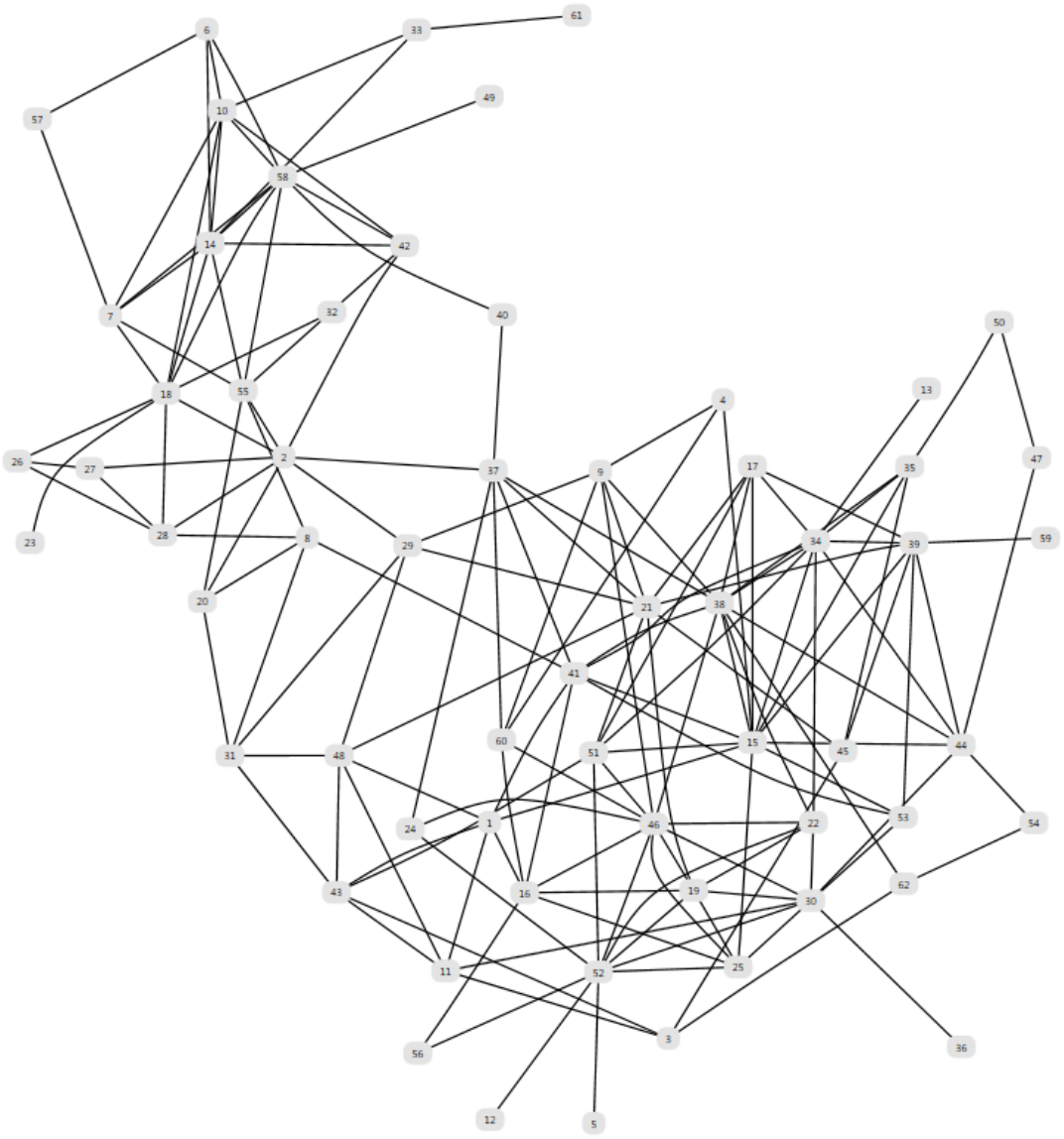
Tabulka 1: Testovací grafy

Pro vykreslení grafu je použit Fruchterman-Reingold layout algoritmus a Force-Scan algoritmus pro eliminaci překrývání vrcholů. Analýza výsledných zjednodušených grafů i původních je provedena pomocí programu Gephi [6].

10.1 Dolphins

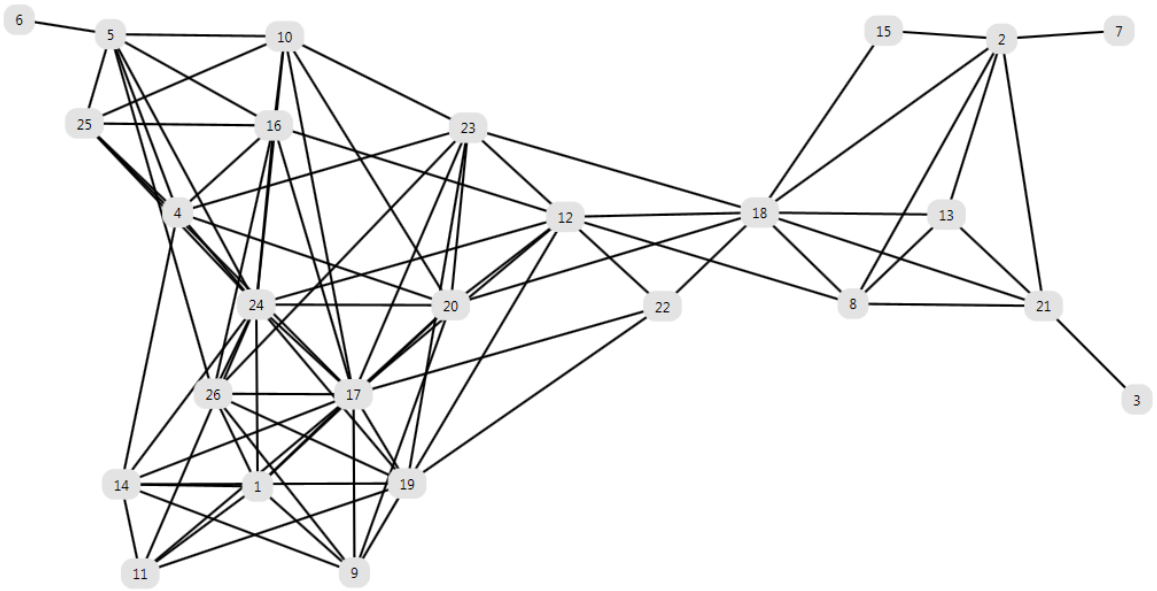
Tento data set [6] představuje sociální síť delfínů skákavých, kteří byli pozorováni při výzkumu jejich chování za účelem dokázat, že se tito tvorové za určitých okolnostech při rozhodování řídí jinými delfíny, se kterými jsou ve skupině [12]. Bylo zjištěno, že stejně jako lidé i delfíni navazují mezi sebou vztahy a podle některých jedinců se ostatní delfíni řídí častěji, než podle ostatních. Vrcholy grafu této sítě představují tyto delfíny a hrany znamenají frekventovanou vazbu mezi dvěma delfíny.

[6] https://gephi.org/



Obrázek 13: Graf sítě delfínů skákavých

Na obr. č. 13 je graf sítě delfínů a na obr. č. 14 je zjednodušený graf této sítě. Délka cesty τ byla nastavena na hodnotu 1 a počet hran byl ponechán bez filtrování.



Obrázek 14: Zjednodušený graf sítě delfínů

Vlastnost	Původní graf	Zjednodušený graf
Počet vrcholů	62	26
Počet hran	159	83
Počet komponent	1	1
Průměrný stupeň	4,954	6,385
Průměr sítě	8	6
Modularita	0,592	0,338
Průměrný shlukovací koeficient	0,297	0,593
Průměrná délka cesty	3,354	2,32

Tabulka 2: Výsledky analýzy sítě delfínů

Tabulka č. 2 porovnává vlastnosti původního a zjednodušeného grafu. Slučováním vrcholů se zvýšil průměrný stupeň, zřejmě kvůli tomu, že metoda sloučila vrcholy s malým stupněm a ponechala centrální vrcholy. Ze stejného důvodu se zmenšil průměr sítě. Naopak se zvýšil průměrný shlukovací koeficient téměř na dvojnásobek původní hodnoty.

10.2 Síť citací DBLP

DBLP [7] označuje bibliografii publikací v oblasti počítačových věd, kterou spravuje německá univerzita Universität Trier. V současné době obsahuje záznamy o 3 328 070 publikací a 1 713 872

[7] http://dblp.uni-trier.de/

autorů z celého světa. Tento data set představuje síť citací mezi publikacemi. Vrcholy grafu sítě reprezentují publikace a každá hrana představuje situaci, kdy je citována v jedné publikaci jiná publikace. Graf je orientovaný. Cílem tohoto experimentu je ověřit metodu na velké síti s nastavením velké hodnoty délky cesty $\tau = 8$.

Vlastnost	Původní graf	Zjednodušený graf
Počet vrcholů	12 591	6 760
Počet hran	49 728	9 595
Počet komponent	40	40
Průměrný stupeň	4,954	2,835
Průměr sítě	8	8
Modularita	0,592	0,691
Průměrný shlukovací koeficient	0,297	0,182
Průměrná délka cesty	3,354	3,555

Tabulka 3: Výsledky analýzy sítě citací DBLP

Výsledný graf je značně zredukovaný. Vlastnosti obou grafů se však liší jen o 10%. Celkový čas výpočtu je 2 minuty 47 sekund.

10.3 High Energy Physics

Tento data set představuje síť vědecké spolupráce mezi autory v oblasti vysokoenergetické teoretické fyziky online knihovny arXiv [8] [10]. Vrcholy grafu sítě představují autory a pokud jsou propojeni hranou, znamená to, že spolupracovali na jednom nebo více článcích. Pokud spolupracuje více lidí na jedné práci, vzniká v síti úplný podgraf. Cílem tohoto experimentu je porovnat výsledky s nastavováním různé délky cesty τ bez použití filtrace hran.

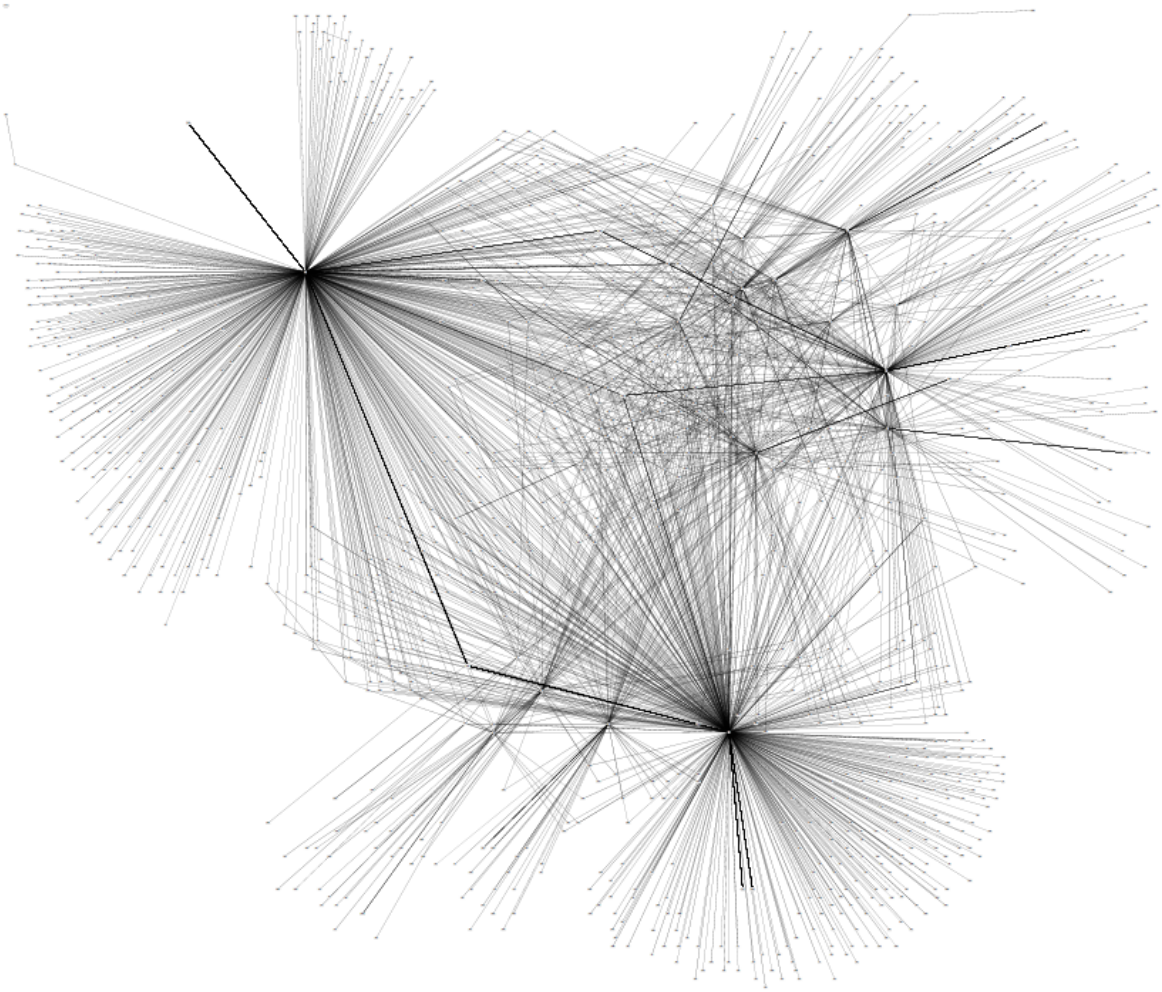
Pro $\tau = 5$:

Vlastnost	Původní graf	Zjednodušený graf
Počet vrcholů	9877	2096
Počet hran	51 946	2578
Počet komponent	429	429
Průměrný stupeň	10,472	3,087
Průměr sítě	13	5
Modularita	0,453	0,487
Průměrný shlukovací koeficient	0,071	0,921
Průměrná délka cesty	3,667	2,516

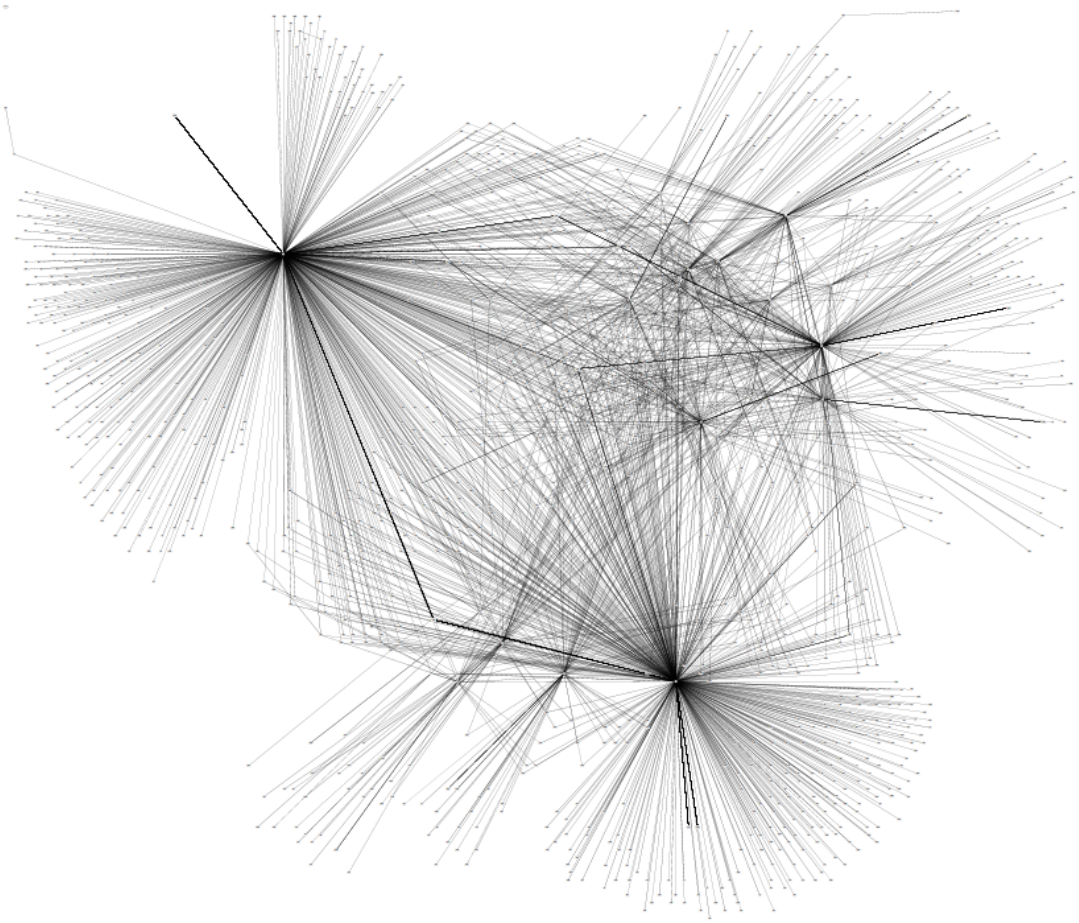
Tabulka 4: Výsledky analýzy sítě High Energy Physics pro $\tau = 5$ bez filtrace hran

[8] arxiv.org

Na následujícím obrázku lze vidět, že vysoká hodnota τ není příliš vhodná pro tento graf. Důsledkem je vytvoření několika málo vrcholů s velmi vysokým stupněm v porovnání s ostatními vrcholy.



Obrázek 15: Zjednodušený graf sítě High Energy Physics pro $\tau = 5$



Obrázek 16: Zjednodušený graf sítě High Energy Physics

Na obrázku 16 je zjednodušený graf této sítě. Délka cesty τ byla nastavena na hodnotu 3 a hran bylo zachováno 100%. Následující tabulka porovnává tento graf s původním:

Vlastnost	Původní graf	Zjednodušený graf
Počet vrcholů	9 877	2 522
Počet hran	59 946	5 498
Počet komponent	429	429
Průměrný stupeň	10,492	5,244
Průměr sítě	13	7
Modularita	0,453	0,42
Průměrný shlukovací koeficient	0,071	0,705
Průměrná délka cesty	3,667	3,105

Tabulka 5: Výsledky analýzy sítě High Energy Physics pro $\tau = 3$

Je vidět, že již při $\tau = 3$ je redukce tohoto grafu značná. Vlastnosti původního grafu se však zcela změnily.

10.4 Síť přátelství Hamsterster

Hamsterer je dnes již zaniklá online sociální síť. Tento data set představuje neorientovanou síť přátelství mezi uživateli této sítě. Cílem prvního experimentu na těchto datech je porovnat kritéria pro určení váhy vrcholů. Parametry τ je nastaveno na hodnotu 3 a počet hran na 80%.

Vlastnost	Původní graf	a	b	c
Počet vrcholů	1858	26	a	c
Počet hran	12534	83	a	c
Počet komponent	23	23	23	23
Průměrný stupeň	13,472	3,543	3,543	3,543
Průměr sítě	13	5	5	5
Modularita	0,457	0,402	0,399	0,398
Průměrný shlukovací koeficient	0,071	0,938	0,938	0,938
Průměrná délka cesty	3,667	2,385	2,385	2,385

Tabulka 6: Výsledky analýzy sítě Hamsterster

- a... průměrná betweenness centralita hran ve shluku
- b... stupeň shluku
- c... počet vrcholů tvořící shluk

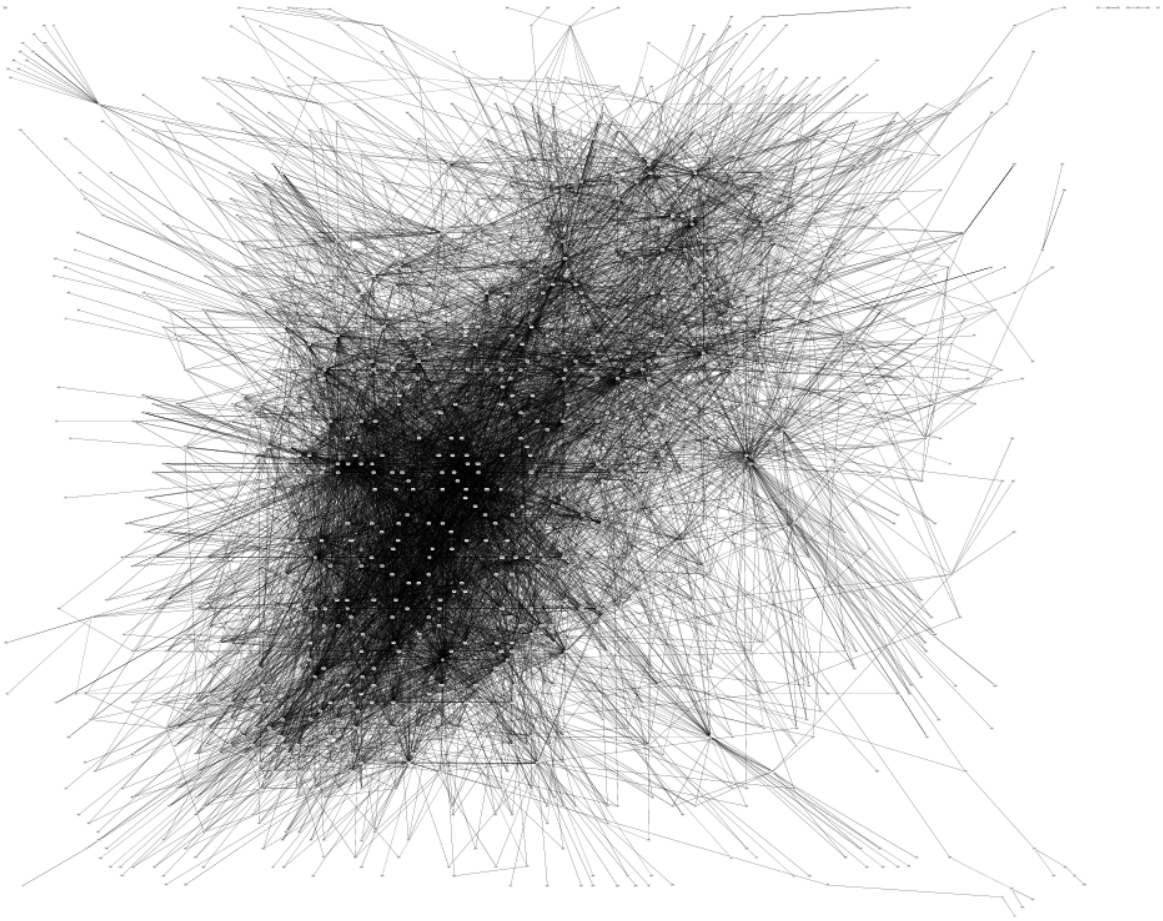
Jak lze vidět, všechny hodnoty zjednodušeného grafu s rozdílnými kritérii pro určení vah vrcholů jsou skoro stejné. Jediný rozdíl je hodnota modularity, která je však zanedbatelná.

Cílem druhého experimentu je porovnat rozdíl mezi vynecháním a použitím metody pro odstranění hran. Délka cesty τ je nastavena na 1. V prvním případě je počet hran ponechán na 100%. V druhém případě je počet hran zredukován na 75% a v třetím již na 50%:

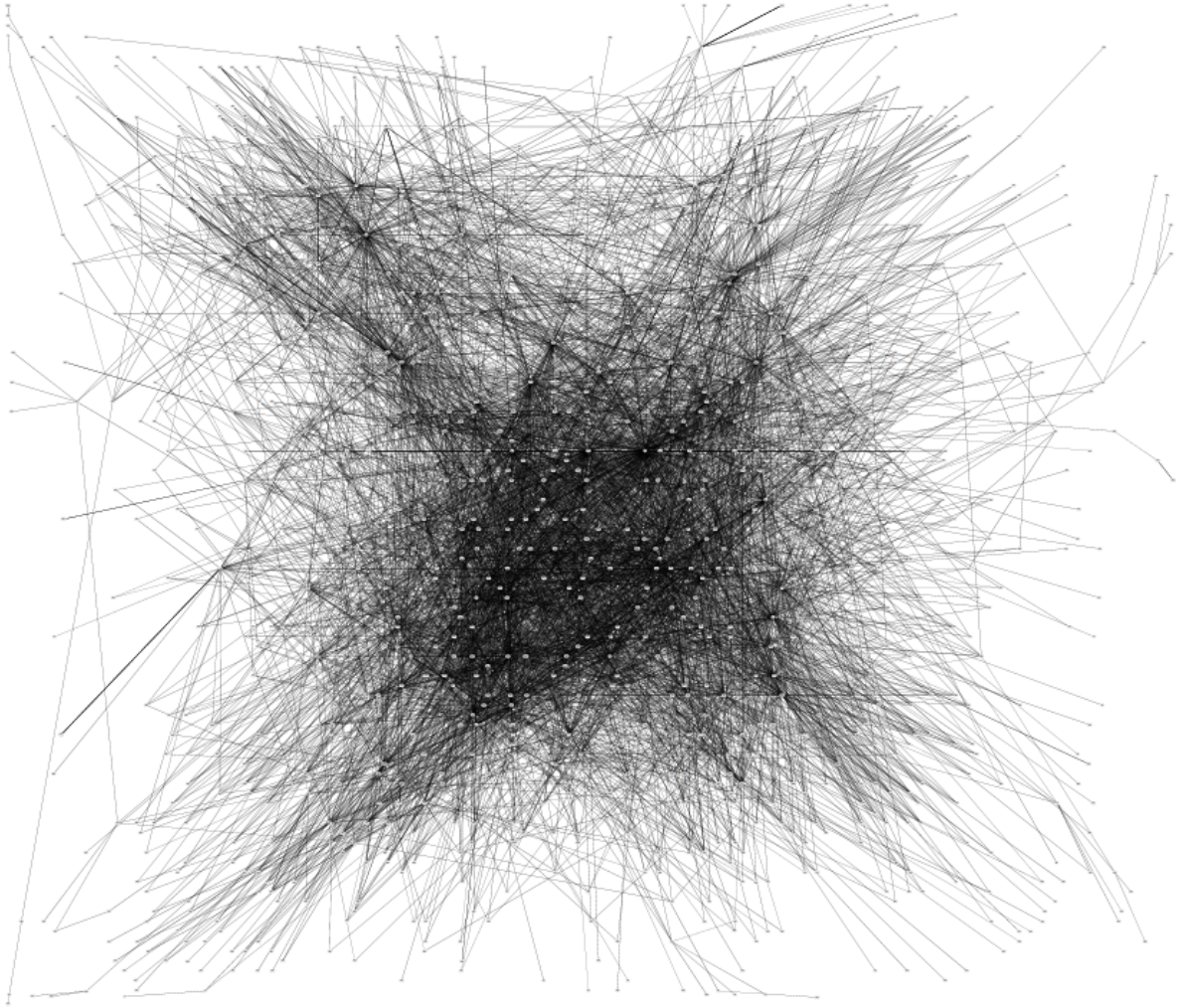
Procentuální počet hran	původní graf	100%	75%	50%
Počet vrcholů	1858	1084	1084	1084
Počet hran	12534	8148	6114	4902
Počet komponent	23	23	23	23
Průměrný stupeň	13,472	15,173	11,385	9,128
Průměr sítě	13	11	11	11
Hustota sítě	0,004	0,014	0,011	0,009
Modularita	0,45	0,38	0,361	0,367
Průměrný shlukovací koeficient	0,071	0,364	0,245	0,215
Průměrná délka cesty	3,667	3,115	3,155	3,219

Tabulka 7: Srovnání výsledků analýzy sítě Hamsterster

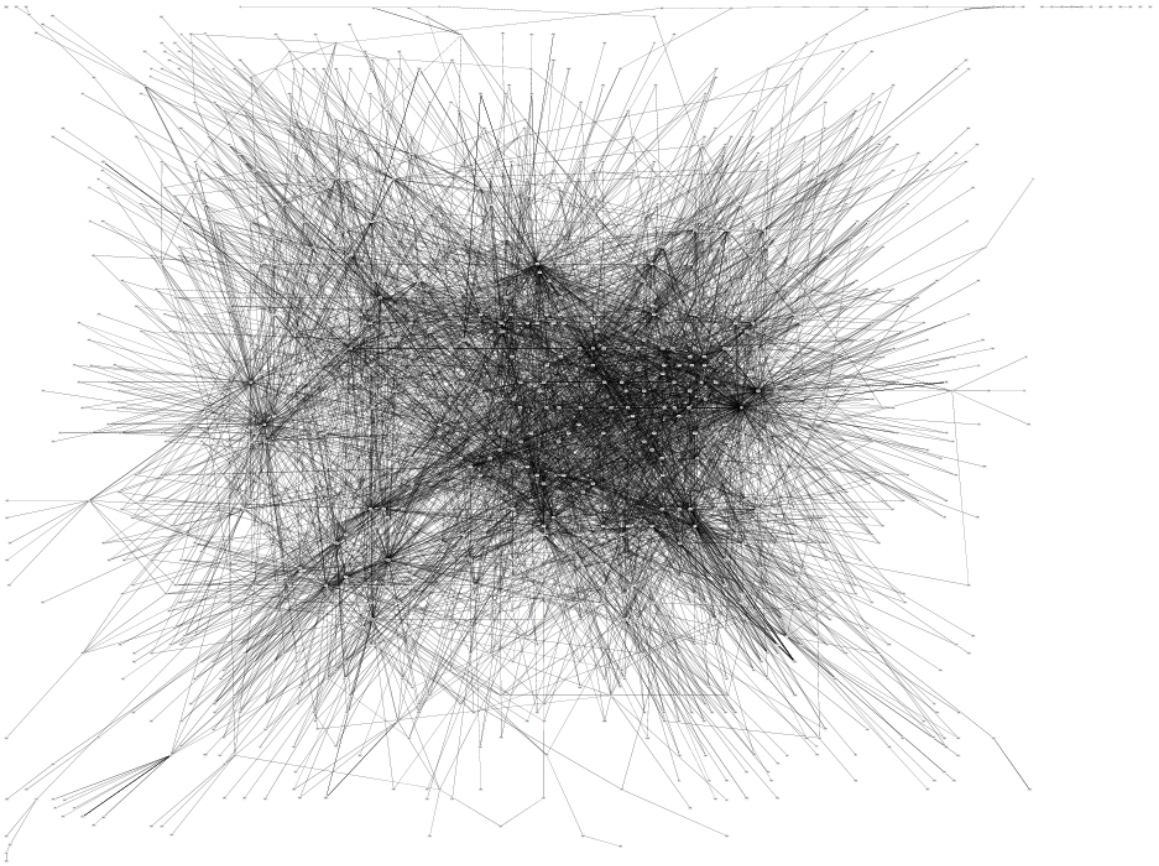
Je zřejmé, že s odstraňováním hran se snižuje průměrný stupeň vrcholů, stejně jako hustota sítě nebo shllukovací koeficient. Se snižováním počtu hran v grafu se však zvyšuje čitelnost tohoto grafu. Zatímco obrázek 17, kdy ještě nebyly odstraněny žádné hrany není moc dobře viditelný, obrázky 18, kdy bylo odstraněno 25% hran a 19 už jsou viditelnější.



Obrázek 17: Graf sítě Hamsterster pro τ=1 bez použití filtrování hran



Obrázek 18: Graf sítě Hamsterster pro τ=1 a odstranění 75% hran



Obrázek 19: Graf sítě Hamsterster pro τ=1 a odstranění 50% hran

10.5 Zhodnocení

Z pozorování usuzuji, že metoda je nejvhodnější pro sítě o velikosti maximálně tisíců vrcholů a desítky tisíc hran, protože větší sítě trvají příliš dlouho. Tato metoda zachovává souvislost komponent, což některé metody, jako je např. random sampling metody nebo edge pruning nezaručují. Podle mého názoru metoda pro filtrování hran vhodně doplňuje navrženou metodu, protože u většiny případů se ukázalo, že hodnota τ by neměla překročit hodnotu 2. Samostatně je tak spíše vhodná pro odstranění vrcholů s nejnižší degree centralitou, což nemění tak razantně vlastnosti sítě. Navíc navržená metoda může být doplněna o detekci a zachování komunit v síti, což může ještě více zpřesnit vytváření zjednodušeného grafu. Na druhou stranu, výpočet betweenness centrality se ukázalo jako velmi časově náročné, což omezuje tuto metodu pro použití na větší grafy. Nutno však zmínit, že existují i rychlejší metody výpočtu betweenness centrality.

Kritéria výpočtu váhy vrcholů byly navrženy třemi způsoby pro účely experimentů. Bylo tak zjištěno, že jsou všechny ekvivalentní co se týče výpočtu vah vrcholů. Proto nedoporučuji používat výpočet pomocí průměrných betweenness centralit shluků, protože to zpomaluje celý proces.

11 Závěr

Cílem této práce bylo seznámit čtenáře s grafy a sociálními sítěmi, nastínit problematiku jejich analýzy a procesem vizualizace. Byly popsány některé algoritmy pro určení vlastností sítí a některé současné metody, které se používají pro získání reprezentativního grafu sítí, které by za normálních okolnostech nebylo možné vizualizovat. Dále byla navržena vlastní metoda, která slučuje vrcholy, které mají nejnižší degree centralitu v síti s vrcholy v určité vzdálenosti. Tato metoda byla doplněna o metodu pro filtrování hran podle jejich betweenness centrality. Tuto vlastnost je však časově náročné určit, zvlášť pro tak velké sítě. Proto je využití této metody omezené.

V praktické části jsem představil implementaci navržené metody a pomocí experimentů nad daty reálných sociálních sítí jsem v poslední části ukázal, že tato metoda se dá používat pro grafy sociálních sítí, které mají i desítky tisíc vrcholů a hran. Dalším krokem může být implementace rychlejšího algoritmu pro výpočet betweenness centrality nebo návrh jiného přístupu, podlé kterého vybírat vrcholy grafu pro slučování například s ohledem na komunitní struktury v grafu.

Literatura

[1] Ulrik Brandes. *A Faster Algorithm for Betweenness Centrality.* Bibliothek der Universität Konstanz, 2001.

[2] Giuseppe Di Battista, Jean-Daniel Fekete, and Huamin Qu. Guest editor's introduction: Special section on the ieee pacific visualization symposium. *IEEE Trans. Visual. Comput. Graphics*, 18(9):1381–1382, 2012.

[3] Robert A Hanneman. *Introduction to social network methods.* University of California, 2005.

[4] Jarke J. Holten, Dannyvan Wijk. Force-directed edge bundling for graph visualization. *Computer Graphics Forum*, 28(3):983–990, 2009.

[5] Yuntao Jia, J. Hoberock, M. Garland, and J. Hart. On the visualization of social and other scale-free networks. *IEEE Trans. Visual. Comput. Graphics*, 14(6):1285–1292, 2008.

[6] Konect, 2016.

[7] Petr Kovár. *Teorie grafu.* 2012.

[8] Valdis Krebs. T n t : The network thinkers: Big (social) data, 2013.

[9] Christos Leskovec, JurijFaloutsos. *Sampling from large graphs.*

[10] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

[11] Shixia Liu, Weiwei Cui, Yingcai Wu, and Mengchen Liu. A survey on information visualization: recent advances and challenges. *The Visual Computer*, 30(12):1373–1393, 2014.

[12] David Lusseau. Evidence for social role in a dolphin social network. *Evolutionary Ecology*, 21(3):357–366, 2006.

[13] Margaret Rouse. What is six degrees of separation? - definition from whatis.com, 2016.

[14] Daniel Thalmann, Jami J Shah, and Qunsheng Peng. *Proceedings, 2009 11th IEEE International Conference on Computer-Aided Design and Computer Graphics.* IEEE Press, 2009.

[15] Mary White. What types of social networks exist?, 2016.

[16] Wikipedia. Scale-free network — Wikipedia, the free encyclopedia, 2016. [Online; accessed 22-February-2016].

[17] Wikipedia. Seven bridges of königsberg — Wikipedia, the free encyclopedia, 2016. [Online; accessed 22-February-2016].

[18] Wikipedia. Social network — Wikipedia, the free encyclopedia, 2016. [Online; accessed 22-February-2016].